Doing mathematics with the APLUSIX-Editor

Denis Bouhineau, Thomas Huguet, Jean-François Nicaud

Doing mathematics and algebra with the APLUSIX-Editor

Denis Bouhineau, Thomas Huguet, Jean-François Nicaud

IMAG-Leibniz, Université de Grenoble
46, rue Félix Viallet, 38031 Grenoble cedex, France
FirstName.LastName@imag.fr
http://aplusix.imag.fr

Abstract. The APLUSIX-Editor is a microworld dedicated to the learning of algebra. First it includes an advanced editor that allows the student to build easily algebraic expressions. Second it reifies the student's reasoning as a linear development of equivalent steps or as a tree. Third it verifies the equivalence between the different steps during student's calculations, and gives indications about the progression of the reasoning.

The first three parts of the paper present the syntactical aspect, the semantic aspect and the strategic aspect of the APLUSIX-Editor. The last parts present experiments conducted with the APLUSIX-Editor.

Introduction

An important part of the mathematical and scientific activities concern two tasks. The first one, we call it formalization or symbolization or 'word problem'. The second one, we call it resolution or 'solving'.

The first task consists in expressing in the form of algebraic expressions a problem often issued from the real life. (*"The book of nature is written in the language of mathematics"* **Galileo**)

This problem has been considered by [Koedinger & al. 1997] and it follows a very interesting tutoring system called "Miss Lindquist designed to carry on a tutorial dialog about symbolization. Miss Lindquist has a tutorial model encoding pedagogical content knowledge in the form of different tutorial strategies that was partially developed by observing an experienced human tutor" (quotation from [Heffernan, Koedinger 2000]).

The second task consists in solving the equation or the system of equations obtained during the formalization of the problem, i.e., finding an algebraic expression equivalent to the initial expression having a certain particular form that is simpler in a sense, people call it 'the solved form'. For example, in the golden ratio problem, the formalization give us that ϕ is solution of E_1: $\phi^2 = \phi + 1$. After resolution, we have E_2: $\phi = {}^1/_2 + {}^{\sqrt{5}}/_2$ or $\phi = {}^1/_2 - {}^{\sqrt{5}}/_2$. From an algebraic point of view, the two previous expressions of ϕ are equivalent ($E_1 \Leftrightarrow E_2$).

It is not rare, in any case in France, that the mathematical activity is reduced to this second part only. It is said, in this case, that the problem is without context. The pure algebraic activity is the only one that appears then.

In the APLUSIX project we are concerned with the second task of mathematical activities, i.e., we are concerned with the resolution of equations, system of equations and inequalities, from step to step, with equivalence between steps.

We place ourselves within a framework where, either the initial problem is expressed in an algebraic form, or the formalization of the problem expressed in natural language is realized as a preliminary, apart from the field of the computer.

Few systems have been defined for this kind of activity, MATHPERT [Beeson 1990], APLUSIX-Tutor (a tutor, ancestor of the APLUSIX-Editor) [Nicaud & al. 1990], but they are seldom used and do not allow any type of activity. In particular, they do not allow a total freedom. Paradox, in classes is more often used the CAS, e.g., Maple [Maple], Mathematica [Mathematica], Derive [Derive], but they lack pedagogical features (they solve a strong problem in one step, they do not provide explanation).

Considering this situation, we thought that there was a place for a new kind of computer systems that will be ergonomic editors of algebraic expressions and reasoning, with a verification mechanism, systems allowing students to make their own calculations and verifying these calculations. Thus, we work on and realized a microworld for mathematics and algebra.

As we have presented it in [Nicaud, Bouhineau on 2001], the mathematical activity maybe decomposed into several levels, the first two ones being the syntax and semantics level. In the present article we will add the strategic level in the description of the APLUSIX-Editor.

At the syntactic level are defined the operators with an arity, a representation, a definition of types for the operand, e.g., we can quote four standard arithmetical operators, digits, brackets… At the syntactic level are also defined the laws of associations which permit to create well formed expressions, which are grouped together according to the types of the expressions (equation, inequalities, system of equations, system of linear equations, polynomials) Also, is defined the notion of sub-expression.

At the semantic level, one have to choose privileged semantics (a reference domain, a denotation) and then is defined a major notion: the notion of equivalence between expressions.

Finally at the strategic level, given a type of particular problem one can give syntactical criteria which defines that the problem is resolved (finished) or not. According to these criteria, rules of progress define a strategy, and can be reifies by indicators of progress.

This article contains five parts. The three first ones concern the three levels (syntactic, semantic and strategic) of the mathematical activity and show which help can be brought by the APLUSIX-Editor to make mathematics. The fourth part reports the results of a first series of experiments made in class.

For legibility, the recent contributions introduced into APLUSIX-Editor will be indicated. They will be described by difference with the system described in [Bouhineau & al. 2001].

A short conclusion contains various points relating to the present and future developments of the APLUSIX-Editor.

1 At the syntactical level

1.1 The edition of algebraic expressions

Until recently, the editors of algebraic expressions were very poor. Most of the computer systems used a 1D editor (even if they used a 2D display). Progress has been made recently, some systems use a 2D editor (e.g., Mathematica [Mathematica], MathType [MathType]). However, their general behaviour is based on text and boxes, not on algebra. For example, with MathType 4, the selection is a text selection; for deleting the denominator of a fraction, one has to copy the numerator, then to select and delete the fraction, last to paste.

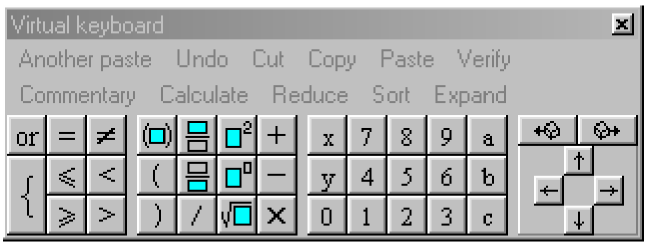

Fig. 1. The virtual keyboard of the APLUSIX-Editor.

The editor part for algebra expressions of the APLUSIX-Editor is 2D (two-dimensions), allowing to modify easily a given expression according to the student's wishes. It uses the usual keyboard and a virtual keyboard (see Fig. 1). The virtual keyboard has been recently introduced, after a first experiments with students having difficulties with mathematics and the usual keyboard.

First, the APLUSIX-Editor includes a mechanism for selecting any well-formed sub-expression, including the situation where the sub-expression has several components. Fig. 2 shows a part of its behaviour. Selection allow making copy/cut/paste and drag&drop in an algebraic way.

Second, three edition modes are defined. The first one is the *structure* mode that takes into account the underlying structured tree of the expressions. This mode allows to insert structured operators with one keystroke, e.g., two parentheses, a fraction, so that there is neither unbalanced parentheses nor fractions without a denominator. The second mode is the *text* mode where a part of an expression is seen as text. This is a non-algebraic mode, but a useful mode for actions like the insertion of digits, variables or unstructured operations like +. The *text* mode allows to insert unbalanced parentheses, which is sometime useful for modifying an expression. The third mode is the *equivalence* mode, a mode in which drag&drop makes calculations, i.e., respects the equivalence of the global expression. The description of this mode is postponed to the next section.

With the *structure* mode, the editor teaches the structure of algebraic expressions. It allows, for example, to make algebraic substitutions with copy and paste, e.g., in order to substitute y–1 to x in 2x+3y=5, one first copies y–1 in the clipboard, then selects x, then pastes. As the paste is performed algebraically, the result is 2(y–1)+3y=5. With this mode, one may transform 8x=2+3x into 8x–3x=2 by doing first a drag&drop of 3x to the left of "=", which produces 8x+3x=2 (where 3x remains selected), second a hit of the "–" key that transforms 3x in –3x.

Third, the editor introduces a question mark ("?") as a special implicit operand where it is necessary in order to have well-formed expression. This operand vanishes automatically when a true operand is typed. This special operand is not considered in the same way than the other symbols: an expression with "?" is understood as not complete, the cursor cannot be just behind or before "?", when it is the case, the cursor is "?", expression like "??" are not possible.

Last, when an expression is syntactically incorrect, e.g. "1/0", "$\sqrt{-1}$" in **R,** "1=2 or 3", the expression is highlighted (in blue for incorrect type, in red for undefined)

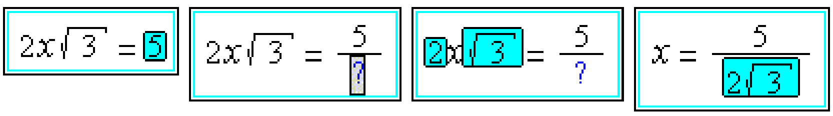

Fig. 2. A selection of 5, a click on the first fraction button of the virtual keyboard, a selection of 2 and $\sqrt{3}$, and a drag&drop to the denominator.

1.2 The edition of algebraic reasoning

Solving a problem in formal algebra consists of searching a solution with the *replacement of equals* inference mode that allows to replace any sub-expression by an equivalent one providing a global expression equivalent to the previous one. Reasoning by equivalence is a very important mechanism. It includes backtrack: in some situations, because of a difficult problem or a lack of knowledge, the student cannot be sure to solve the problem a direct way and needs to go back to a previous step in order to try another way. A good presentation and edition of a reasoning process by equivalence, including backtrack, is necessary.

CAS generally mix algebraic display and calculations with text processing. So they fulfil, in a sense, the above features but in a way asking the student to be in charge of the whole presentation of the reasoning. An editor of algebraic reasoning has to be in charge of the presentation.

In the APLUSIX-Editor, the edition of algebraic reasoning has been realised for reasoning by equivalence. The presentation is made of boxes that contain the expressions and links that indicate the transitions and the equivalence. An example with backtrack is shown in Fig. 3.

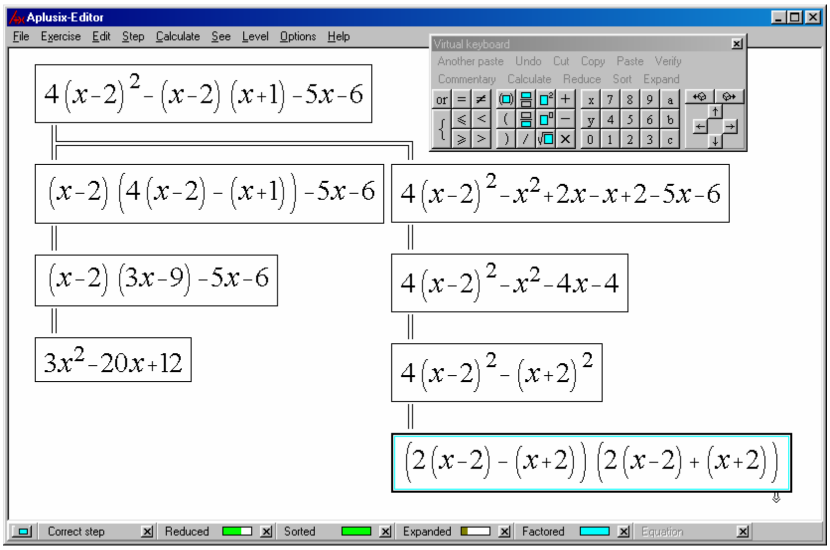

Fig. 3. The student tried a partial factoring of x-2 but did not get a common factor. Then (s)he expanded the expression but did not get an identity. At that moment, (s)he decided to backtrack to the first step in order to try a partial expansion.

Recent development in the construction of the APLUSIX-Editor have introduce three kinds of link between expressions: Simple line, between expression when equivalence between expression is not calculated, double line (like in Fig. 3) for polynomial, when equality between polynomials is calculated, and double arrow (like in Fig 4.) for equation, inequalities, system of equation when equivalence between expression is calculated.

In next section, we will define how the last two kind of link will be enriched to express the semantic of the reasoning process.

2 At the semantic level

2.1 The verification of the students' calculations

Two algebraic expressions are equivalent if and only if they have the same semantics, called denotation. For example, over the real numbers, the two expressions (x-1)(x+1)-3 and (x-2)(x+2) are equivalent because the two functions x $\rightarrow$ (x-1)(x+1)-3 and x $\rightarrow$ (x-2)(x+2), from **R** to **R**, are identical. This property allows to verify calculations without knowing what transformation has been done. Such verification mechanism in an editor of algebraic reasoning is very important for the student because it verifies his/her calculations.

As far as we know, there are no computer systems doing that, because all the systems for formal algebra (CAS or educational systems) are action driven, the action being performed by the computer, so the student do not input his/her own results. However, it is very important for the student to face situations where his/her errors are apparent. This favours learning by constructing more adequate knowledge.

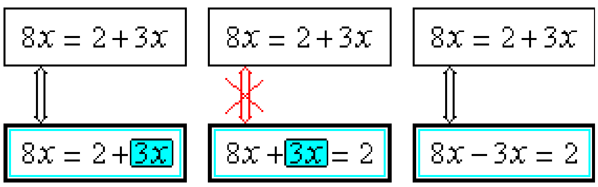

Fig. 4. (1) The equation is duplicated and 3x is selected. (2) A drag&drop of 3x: The equations are no more equivalent, the link is crossed in red. (3) A hit of "–": The equations are now equivalent, the equivalence link is restored.

In order to verify the students' calculations, the APLUSIX-Editor calculates the equivalence between the expressions. Currently, these calculations are implemented on the field of real numbers for polynomial expressions (n variables, degree p), for systems of linear equations (n variables, p equations), and for equations and inequalities (degree less or equal 2). At the present time, we use approximate calculations. They will be soon replaced by exact calculations for rational numbers. When the expressions are not equivalent, the system crosses the link between them (see Fig. 4). A red cross signify that expressions are not equivalent, a blue cross is used when the expression is not complete.

In fact, there are two modes of evaluation for the equivalence. In the first mode, the calculation of the equivalence is done permanently. In that situation, links are drawn as double links (double line and double arrow). At each stroke of key made by the student, the equivalence between the current step and the previous one (in the reasoning tree) is evaluated and the arrow is updated. In the second mode, no calculation is done in permanence. In that situation, links as drawn as a simple line. The choice between these two modes belongs to the user. However, when the second mode is chosen, the user can ask for one particular evaluation of equivalence. These two modes, and the evaluation 'when asked' are recent introductions in the APLUSIX-Editor.

2.2 Drag&drop in *equivalence* mode

In the previous section, we have mentioned an *equivalence* mode for drag&drop. Concerning the drag&and drop of 3x in the expression 8x=2+3x, that mode would permit to transforms directly that expression in 8x–3x=2. The drag&drop in the equivalence mode is not yet implemented. More examples of drag&drop in an *equivalence* mode are given in table 1. They are based on factorisations or reductions.

Expression	Selected sub-expression	Place of the drop	Result	Type of action
$3x^2-1+x^2$	x^2	over $3x^2$	$4x^2-1$	Reduction
$(y-1)(x+x^2)$	first occurrence of x	Between)(	$(y-1)x(1+x)$	Factorisation
$(xy)^2$	x	Before (	x^2y^2	Factorisation
$\sqrt{4+x}$	4	Before sqrt	$2\sqrt{1+\frac{x}{4}}$	Factorisation
$\sqrt{2+x}$	2	Before sqrt	$\sqrt{2}\sqrt{1+\frac{x}{2}}$	Factorisation
$\frac{x}{x+y}$	first occurrence of x	Denominator	$\frac{1}{1+\frac{y}{x}}$	Double factorisation and reduction

Table 1. Examples of equivalent drag&drop with basic operators.

2.2 Direct manipulation for the calculations and Dynamic Algebra

The drag&drop introduced in the previous section for the *equivalence* mode is typically a kind of direct manipulation, it concerns expressions. We are thinking also about direct manipulation for the calculations. This would lead to the comportment shown Fig. 5. This form of direct manipulation is not yet implemented. We hope, when implemented, it will introduce a new kind of algebra 'dynamic algebra', in the same way as 'dynamic geometry' as been introduced with Cabri [Cabri].

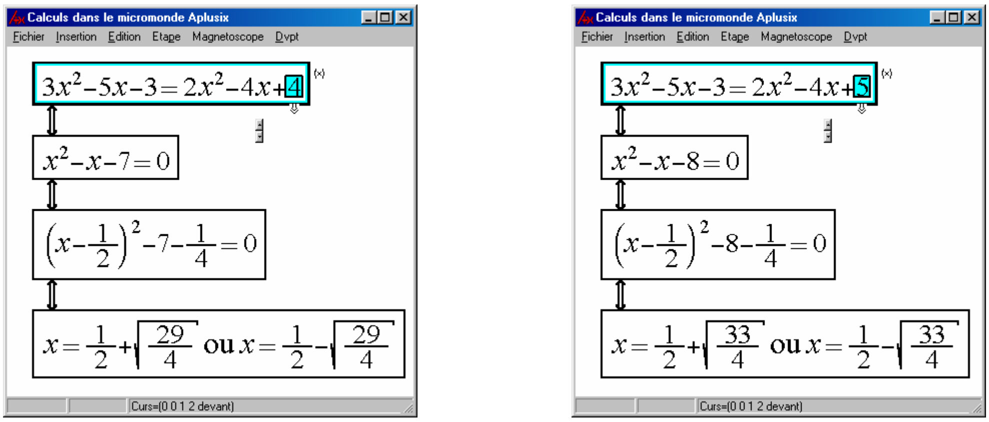

Figure 5. Direct manipulation for the calculus, the selected digit 4 is transformed into 5 and the calculus changed according to that new value.

3 At the strategic level

3.1 Epistemic feedback

Microworlds are based on feedback. The first level of feedback consists of displaying the objects when they are modified. The second level consists of providing conceptual or semantic information [Hollan & al., 1984].

Concerning formal algebra, an important semantic feedback is the *correctness* of the calculations. Because it is linked to the denotation, we have presented this issue in the previous section.

Other interesting feedbacks are indications about general concepts having an important role for many problem types. For example, the *reduced* concept is important because we quite always reduce expressions that are not reduced; the *sort* concept is interesting because there are many situations where an order on commutable expressions is preferred.

A very interesting feedback is the *solved* concept: telling the student whether the problem is solved or not. In fact, this concept depends on the problem type and a microworld for algebra is devoted to any sort of problem types. Consequently, a *solved* feedback can be implemented for some problem types but a general *solved* feedback cannot be implemented.

3.2 Indicators

Indicators have been recently introduced in APLUSIX-Editor. There have not been any experiment done with them yet, but are directly connected to the experiments already achieved. During these experiments, one of the most frequently asked question was "Is it finished?" This introduces the notion of strategy. They are associated with the following concepts: factored, reduced, solved, sorted, expanded. These concepts are detailed below. In Fig 3, the indicators associated to these concepts are represented at the bottom of the window of the application.

The indicator linked to the factored concept have been implemented like a gauge indicating the ratio between the number of prime polynomials that are factored over the number of prime polynomials of the totally factored form. In fact, the implemented factored concept is limited to polynomial expressions of one variable. This concept depends on the set of numbers used, **R** or **C**, (prime polynomials being first degree polynomials over the complex numbers and first degree polynomials or second degree polynomials having a discriminant less than 0 over the real numbers).

The *reduced* concept is based on a set R of rewrite rules having a reduction status in a wide sense. Examples of reduction rules are:

$A+0 \to A$ $\quad -(-A) \to A$ $\quad AA \to A^2$ $\quad \frac{a}{ab} \to \frac{1}{b}$

In order to have a strong concept, arithmetical calculations (e.g., replacing 3+4 by 7) and arithmetical simplifications of fractions (e.g., replacing $\frac{12}{15}$ by $\frac{4}{5}$) must be seen as reductions, so is the grouping of like terms in a sum (e.g., replacing $2x^2+3x^2$ by $5x^2$).

In that framework, an expression E is reduced if there is no reduction rule applicable to a sub-expression of E.

The indicator associated with that concept has been implemented in the APLUSIX-Editor with a gauge depending on the number of reduction rules that are applicable.

We implemented in the system a *sorted* concept in the same way, using a set of sort rules. Among them are: commute A and B in a product if A is a number and B is not; commute two monomials A and B of the same variable in a sum if the degree of A is less that the degree of B.

We implemented an *expanded* concept based on the number of parentheses in the expression.

We have introduced a special indicator, called ‘Equation’ for equation, inequalities and system of equations, for the ‘solved’ concept. The *Equation* gauge indicates a degree of progression towards a solved form for equations, inequalities and systems of linear equations.

The solved forms of some other problem types can be translated in terms of the above concepts. For example, *Factor a polynomial expression* may be seen as getting an equivalent form that is *factored* and *Reduced*; *Expand a polynomial expression* may be seen as getting an equivalent form that is *Expanded* and *Reduced.*

4 First experiments of the APLUSIX-Editor

Our first tests took place in autumn 2001, inside a regular classroom, and in January 2002, as a controlled experiment. In both cases, we worked with beginners in algebra (14 - 15 years old). The version we used for these experiments had no indicator and no command.

4.1 Using the APLUSIX-Editor in a class

During the whole month of December 2001, the 18 students of a class used the system several times a week. The class was a special class with many students having deep difficulties with mathematics. The teacher of this class was involved in the project. The students started learning expansions, simplifications and factorisations of simple expressions, and resolution of simple equations with the APLUSIX-Editor. Some students worked alone with the computer, others worked in groups of two. Most of the students needed just a few minutes to appropriate the software, even those who did not have an important experience of the computer. Some of them acquired a great mastery of the drag&drop function, using it as well to copy some elements of an expression inside a new step, as to move parts of an expression to transform it, in particular in order to move a term from a side of an equation to the other. Figure 6 gives examples of typical resolutions.

$$4x-9=12x+13$$
$$\Updownarrow$$
$$4x-12x=13+9$$
$$\Updownarrow$$
$$-8x=22$$
$$\Updownarrow$$
$$x=-\frac{11}{4}$$

$$(3x+5)\ (5x+9)=0$$
$$\Updownarrow$$
$$3x+5=0 \text{ or } 5x+9=0$$
$$\Updownarrow$$
$$x=-\frac{5}{3} \text{ or } x=-\frac{9}{5}$$

Figure 6. Typical resolution, on the left, of a first-degree equation and on the right, of a "product equation".

For many students, what was usually opaque in algebra regained interest. Some of them, who generally did not listen, all of a sudden began to ask questions. From passive, they became active. They all enjoyed the equivalence checking. This functionality moved the exchanges between the teacher and the students: "*Is it right?*" became "*Why is it wrong?*" This shifts to what a problem really is.

All the students took pleasure in going to the computer lab and often asked to go there again. When working with the APLUSIX-Editor, they solved more exercises than usually and exercises that are more difficult.

4.2 A controlled experiment

We led a controlled experiment of the APLUSIX-Editor on the 9th and 16th of January 2002, with a group of 8 volunteers, coming from different classes. They worked apart from the normal classroom.

Our objective was to check our first impressions by measuring the realised progresses and analysing interaction. Table 2 shows the experimentation plan.

Day	Phase	Duration
9 January	Pre-test on paper	30 minutes
	Session with the APLUSIX-Editor	1 h 30
16 January	Session with the APLUSIX-Editor	1 h 30
	Post-test on paper	30 minutes
	Questionnaire	15 minutes

Table 2. The experimentation plan.

The first session dealt with the basic notions of equation solving (carry out the same operation – add, divide… – on both sides of the equal sign). The second one emphasized on the transformation rule that allows to move an additive term from a side of an equation to the other, if its sign is changed. The teacher began by briefly presenting the lesson and drove both sessions. Then the students worked alone with the system, which was automatically supplied with exercises previously placed in a file. A large variety of exercises was proposed. In that context, the students were able to evolve according to their own speed. The teacher's role was to answer to the student's questions. He intervened sometimes spontaneously. An observer was in charge of systematically taking notes on the exchanges between the teacher and the students.

4.3 Study of the pre-test and the post-test

From the pre-test to the post-test, the average of the group increased from 4.2 out of 10 to 7.9 while the standard deviation decreased from 3.4 to 2.8. Two good students progressed just a little; three students multiplied their note by three or four. Although these results have to be interpreted cautiously, because of the small effective and the imprecision of any assessment, we think that they testify of an important progression.

The students' learning did not confine to the transformation rules. Some students, who often deconstructed their calculations, presented their calculations during the post-test as expected by the teacher, see figure 7. A few students who already had some mastery began to mentally apply some rules.

	Pre-test	**Post-test**	**Comment**
Student 1	$14x-17=9x+6$ $x-17=9x+6 \ :14$ $\frac{x}{x}-17=9+6:14$ $x = \frac{15+17}{14}$ $x=\frac{32}{14}$	$14x-17=9x+6$ $14x-17-9x=6$ $5x=6+17$ $5x=23$ $x=\frac{23}{5}$	*The learning of transformation rules and the structuring of the calculation presentation.*
Student 2	$1=-4x+10$ $1+4x=10,$ $4x=10-1.$ $4x=9.$ $x=\frac{4}{9}.$	$1=-4x+10$ $4x=9.$ $x=\frac{9}{4}$	*During the post-test, student 2 no more needed to write some intermediate calculation steps.*

Figure 7. Some improvements of student behaviour between the two tests.

4.4 Protocols analysis

The APLUSIX-Editor saves the user's actions (keystrokes, drag&drop...). The protocols obtained in the two sessions have been analyzed. We have extracted statistical information and replayed resolutions (the system allows to replay the interactions memorized in protocols like a video recorder).

Table 3 indicates the minimal times (the fastest student), maximal time (the slowest student) and average time (group's average) taken by students to solve some exercises.

We notice an increase of time when an exercise has something new (e.g., from 1a to 2a), a decrease of time after the software appropriation (e.g., from 4a to 10a) and a decrease of time between the first and the second session for similar exercises (e.g., from 3a to 2b).

	Exercise number	**Equation**	**Time (in second)**		
			Maximum	**Average**	**Minimum**
Session 1	1a	8x-7=13x+3	1094,2	371,1	129,6
	2a	7(2x+4)=-x-5	1743,5	607,9	182,9
	3a	-5x+7=2(-8x+9)	747,4	326,1	154,6
	4a	x+1=7x	955,8	297,3	77,1
	10a	10x+8+17x-13=6x+3	229,5	160,0	78,7

Session 2	1b	1+4x=x-3	359,2	265,0	162,1
	2b	3x+7=5(x+2)	248,8	172,0	108,7
	3b	2-2x=5x+3	376,7	136,2	76,7
	4b	2(x-1)=3(2+x)	388,5	160,8	79,3

Table 3. Times taken by students to solve some equations.

With the protocols, it is possible to follow all the story of each resolution. We can see what the student tried, what worked and which algebraic points are problematical.

Our analysis showed the important role of the equivalence checking functionality. Here is an example. A student made progress during the two sessions, beginning by adding the same expression to both sides of the equation/inequality, then moving an additive expression to the other side. He got confidence in this procedure and applied it rather quickly. In the middle of the second session, he had to solve the inequality 4x–1–(2–3x)<3x–5(2–x)+1. He first expanded, typing 4x–1–2+3x<3x–10+5x+1. At the end of this input, the system indicated equivalence. The student created then a new step and typed regularly 4x+3x+3x+5x<1–10+2+1. At the end of this input, the system indicated non-equivalence. After 10 seconds without action, the students clicked before the second 3x, deleted the "+" and typed a "–". Again, the system indicated non-equivalence. After 17 seconds without action, the students changed the sign of 5x and got the equivalence. It seems clear that the student had an idea of what could be wrong, because he went directly to the problem. What would have happened in a paper/pencil context? The student would just have continued the resolution with a wrong expression and would not have reinforced his learning of the correct procedure. On another exercise, later, the student began to combine several transformations, which requires many mental calculations. When he got a non-equivalence feedback, he deleted the entire expression and replaced it with another built with less combination, and without error.

4.5 The observer's observations

During the second session, the observer noticed 28 exchanges between the teacher and the students: 13 came from a "*Is it finished?*" question and 15 dealt with algebraic difficulties, often beginning by "*Why is it not working?*" Note that the system was not equipped with indicators.

The teacher never gave directly the solutions of the 15 difficulties that concerned calculations over fractions and negative integers. He always tried to let the students think about an isomorphic situation or to give the minimal elements that could allow the students to retrieve the underlying arithmetic fundamental rules.

From a pedagogical point of view, all these difficulties dealt less with the current didactic situation than with older and unaccomplished learning situations, but this is not sufficient to explain why the students were asking for help on several mathematical aspects and not on others. An epistemic regard can give us elements of response to that fact.

The students never asked for help about how to correctly modify an equation after a mistake, about what to do next at a certain state of their reasoning, or about the methods for solving equations, although they did not already know all of the aspects of the lesson and were not applying a mastered knowledge. As we saw in the protocols, the students frequently encountered difficulties, employing erroneous algebraic transformation rules or trying something completely new respectively to their previous behaviour.

In brief, as strange as it can seem, the themes of all the students' calls to the teacher never related, directly or not, to the algebraic equivalence, although it was indirectly the key of all of their training. Actually, this is not so paradoxical, if we compare the current feedbacks of the APLUSIX-Editor (e. g., algebraic edition and equivalence checking functionality) and the content of the exchanges between the students and the teacher (e. g., form of the solution and arithmetic). A contrary correspondence can be established between them: where adequate feedbacks are provided, the exchanges tend to disappear. This leads us to say that the choices made in the design of the equivalence checking functionality are didactically completely pertinent.

The students were able to develop their own knowledge by working with the APLUSIX-Editor. The algebraic techniques they used practically at the end of the second session were just what the teacher expected. Thus, by referring to Brousseau's theory of didactical situations [Brousseau 1997], we can say that, in a certain way, the APLUSIX-Editor contains characteristics that allow proposing to students situations with a-didactical features. The precise studies of these characteristics and of the ways to make the system as a-didactic as possible are two of our major goals.

Reading the observer's notes, we became aware of the need of a feedback concerning the state of the problem. We did not implement such kind of feedback in the first version. This need led us to define and implement in the second version the indicators described in section 3.

4.6 Students' answers to the questionnaire

A part of the questionnaire was devoted to interest (compared to paper), difficulty, and usefulness. The results are:

Less interesting:	0	As interesting as:	1	More interesting:	7		
Very difficult:	0	Rather difficult:	3	Rather easy:	4	Very easy:	1
Entirely useless:	0	Rather useless:	0	Useful enough:	0	Very useful:	8

In their explanations, the students wrote that it is simpler to correct by typing on the keyboard and that it is very pleasant that the APLUSIX-Editor verifies what they do.

One of the requests we found in the questionnaire was to use the APLUSIX-Editor every week during the help-time (a moment in the week devoted to help students to solve their difficulties in French, History and Geography and Mathematics, driven by three teachers in a classical classroom).

Recent and future works

The first couple of tests of the APLUSIX-Editor is a real success and suggests that the system may fulfil a gap in educational software for algebra, as we thought.

There are many prospects for the APLUSIX-Editor.

First, we want to distribute it wildly. We distribute a free version, limited in time, in several languages (http://aplusix.imag.fr).

Second, we will build tools for analysing the protocols so that researchers in maths education and teachers may study the work of their students. We are currently involved in a project for the analysis of algebra learning, the construction of student models and the elaboration of strategy for algebra learning.. That project includes researchers in psychology, mathematics education and machine learning. It is funded by the French Ministry of research. The APLUSIX-Editor is already equipped with a mechanism for interaction recording and replaying.

Third, we will make it evolve towards an Ed-CAS (Educational Computer Algebra Systems), another new kind of system for algebra that we define as an algebraic editor with graph functionalities and commands like the ones we have in a CAS. We have already introduced recently few commands, you can see on Fig 1.: calculate, reduce, sort, expand. They are based on rewriting rules, as are some indicators.

Fourth, we will add a tutor over the system, for some problem types, and reuse, for that, our previous work on the APLUSIX-Tutor.

References

[Beeson 1990] Beeson, M. (1990). Mathpert, a computerized learning environment for Algebra, Trigonometry and Calculus, Journal of Artificial Intelligence in Education, p. 65-76.

[Bouhineau & al. 2001] Bouhineau D. Nicaud J.F., Pavard X., Sander E. (2001). A Microworld For Helping Students To Learn Algebra. Proceeding of ICTMT5, Austria.

[Brousseau 1997] Brousseau G. (1997), *Theory of Didactical Situations in Mathematics,* Kluwer Academic Publishers.

[Cabri] Cabri http://www-cabri.imag.fr/

[Derive] Derive http://education.ti.com/product/software/derive/features/features.html

[Heffernan, Koedinger 2000] Heffernan, N. T., & Koedinger, K. R. *Building a 3rd generation ITS for symbolization: Adding a Tutorial Model with Multiple Tutorial Strategies*. Intelligent Tutoring Systems (2000) Conference Workshop entitled "Learning Algebra with the computer, a transdisciplinary workshop".

[Hollan & al. 1984] Hollan J. D., Hutchins E. L., Weitzman L. (1984). *STEAMER: an interactive inspectable simulation-based training system*. AI Magazine, vol 5, n° 2.

[Koedinger & al. 1997] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). *Intelligent tutoring goes to school in the big city*. Int. Journal of Artificial Intelligence in Education, 8.

[Maple] Maple http://www.maplesoft.com

[Mathematica] Mathematica http://www.wolfram.com

[MathType] MathType http://www.mathtype.com/msee

[MissLindquist] Miss Lindquist http://www.algebratutor.org

[Nicaud & al. 1990]. Nicaud J.F., Aubertin C., Nguyen-Xuan A., Saïdi M., Wach P. (1990). APLUSIX: a learning environment for student acquisition of strategic knowledge in algebra. Proceedings of PRICAI'90. Nagoya.

[Nicaud, Bouhineau 2001] Nicaud J.F., Bouhineau D. (2001). Syntax and semantics in algebra. Proceedings of the 12th ICMI Study Conference. The University of Melbourne.